**Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики.**

Факультет Программной Инженерии и Компьютерной Техники



Лабораторная работа №2

предмет «Программирование» вариант 311233

ФИО: Фам Мань Туан

Группа: P3112

Преподаватель: Максимов Андрей Николаевич

Перцев Тимофей Сергеевич

**Санкт-Петербург 2020 г.**

Задание:

На основе базового класса Pokemon написать свои классы для заданных видов покемонов. Каждый вид покемона должен иметь один или два типа и стандартные базовые характеристики:

* очки здоровья (HP)
* атака (attack)
* защита (defense)
* специальная атака (special attack)
* специальная защита (special defense)
* скорость (speed)

Классы покемонов должны наследоваться в соответствии с цепочкой эволюции покемонов. На основе базовых классов PhysicalMove, SpecialMove и StatusMove реализовать свои классы для заданных видов атак.

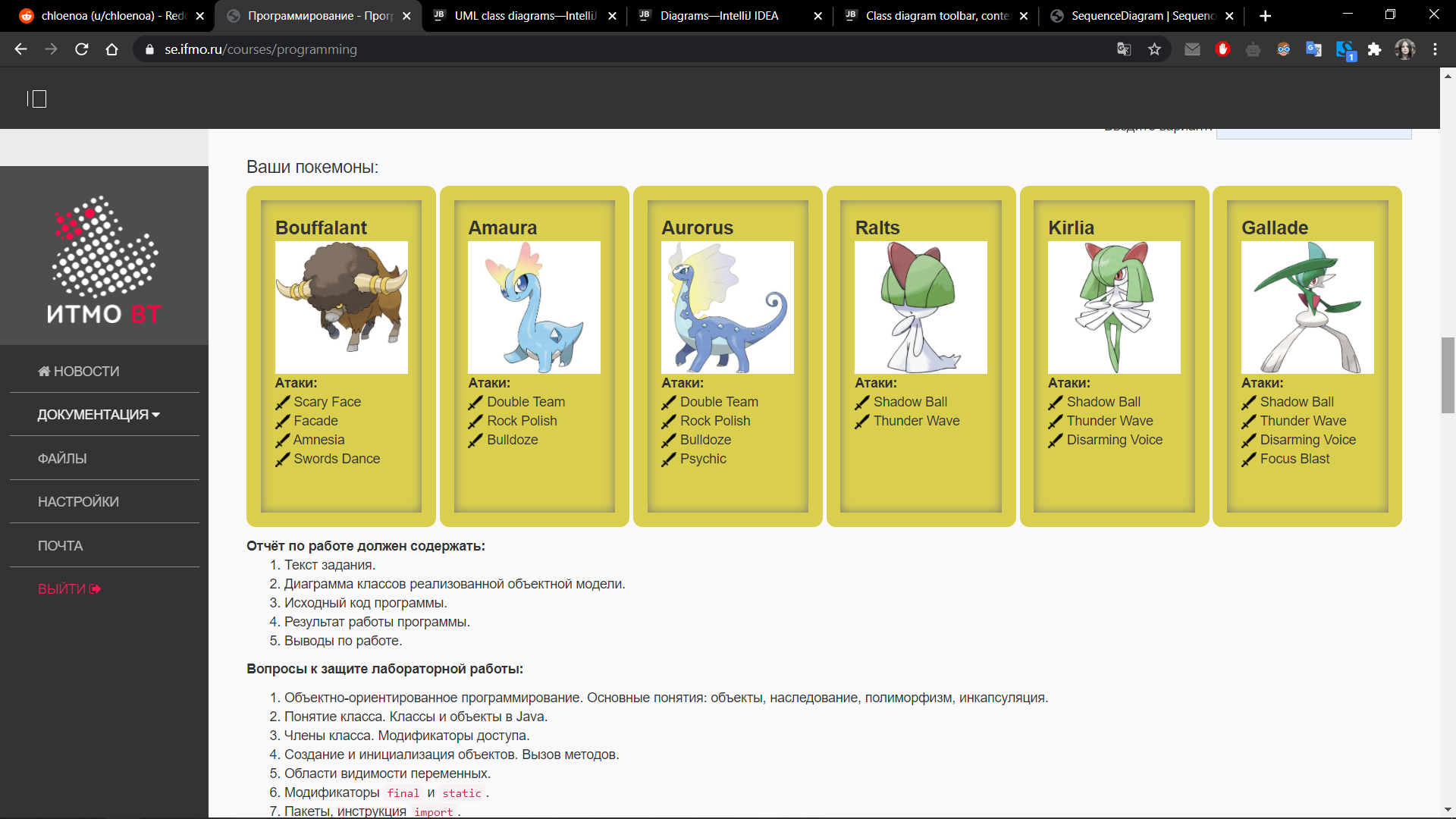
Атака должна иметь стандартные тип, силу (power) и точность (accuracy). Должны быть реализованы стандартные эффекты атаки. Назначить каждому виду покемонов атаки в соответствии с вариантом. Уровень покемона выбирается минимально необходимым для всех реализованных атак.

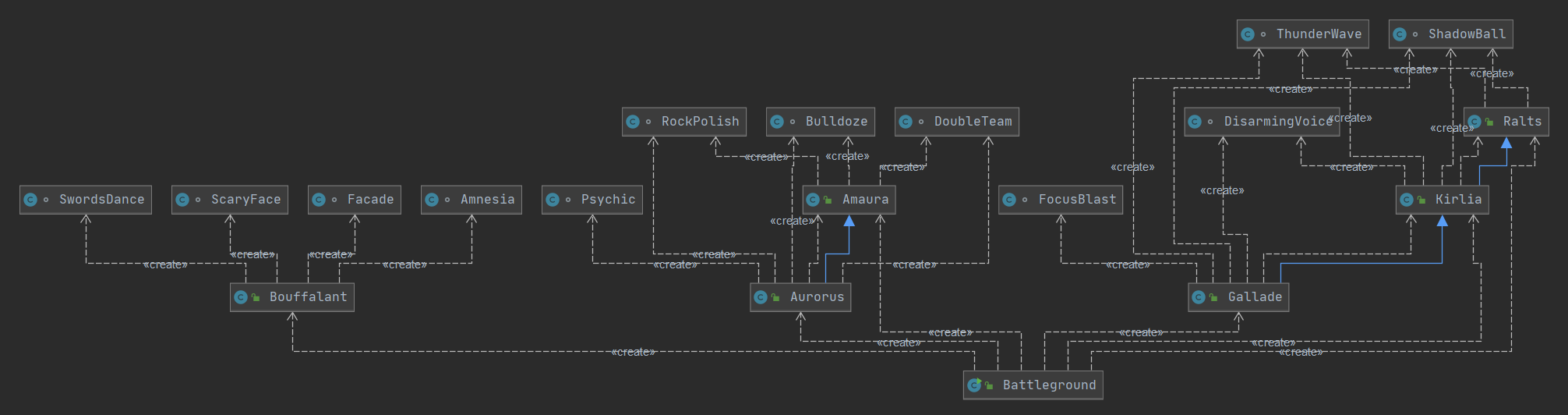
Используя класс симуляции боя Battle, создать 2 команды покемонов (каждый покемон должен иметь имя) и запустить бой.

Базовые классы и симулятор сражения находятся в [jar-архиве](https://se.ifmo.ru/documents/10180/660917/Pokemon.jar/a7ce60af-6ee6-47d0-a95e-e5ed9a697bd2) (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате javadoc - [здесь](https://se.ifmo.ru/~tony/doc/).

Информацию о покемонах, цепочках эволюции и атаках можно найти на сайтах <http://poke-universe.ru>, <http://pokemondb.net>,<http://veekun.com/dex/pokemon>

Покемоны



Диаграмма класс

Исходный код

Battleground.java

import ru.ifmo.se.pokemon.\*;  
  
//zoomer zone  
  
public class Battleground {  
 public static void main(String[] args) {  
 Battle field = new Battle();  
 field.addAlly(new Bouffalant("Travis",1));  
 field.addAlly(new Amaura("Kanye",3));  
 field.addAlly(new Aurorus("Frank",3));  
 field.addFoe(new Ralts("Kendrick",2));  
 field.addFoe(new Kirlia("Tyler",3));  
 field.addFoe(new Gallade("Abel",2));  
 field.go();  
 }  
}

Amaura.java

import ru.ifmo.se.pokemon.\*;  
  
public class Amaura extends Pokemon {  
 public Amaura(String name, int level) {  
 super(name, level);  
 setStats(77,59,50,67,63,46);  
 setType(Type.*ROCK*, Type.*ICE*);  
 setMove(new DoubleTeam(), new RockPolish(), new Bulldoze());  
 }  
}

Aurorus.java

import ru.ifmo.se.pokemon.\*;  
  
public class Aurorus extends Amaura {  
 public Aurorus(String name, int level) {  
 super(name, level);  
 setStats(123,77,72,99,92,58);  
 setType(Type.*ROCK*, Type.*ICE*);  
 setMove(new DoubleTeam(), new RockPolish(), new Bulldoze(), new Psychic());  
 }  
}

Bouffalant.java

import ru.ifmo.se.pokemon.\*;  
  
public class Bouffalant extends Pokemon {  
 public Bouffalant(String name, int level) {  
 super(name, level);  
 setStats(95,110,95,40,95,55);  
 setType(Type.NORMAL);  
 setMove(new ScaryFace(), new Facade(), new Amnesia(), new SwordsDance());  
 }  
}

Gallade.java

import ru.ifmo.se.pokemon.\*;  
  
public class Gallade extends Kirlia {  
 public Gallade(String name, int level) {  
 super(name, level);  
 setStats(68,125,65,65,115,80);  
 setType(Type.*PSYCHIC*, Type.*FIGHTING*);  
 setMove(new ShadowBall(), new ThunderWave(), new DisarmingVoice(), new FocusBlast());  
 }  
}

Kirlia.java

import ru.ifmo.se.pokemon.\*;  
  
public class Kirlia extends Ralts {  
 public Kirlia(String name, int level) {  
 super(name, level);  
 setStats(38,35,35,65,55,50);  
 setType(Type.*PSYCHIC*, Type.*FAIRY*);  
 setMove(new ShadowBall(), new ThunderWave(), new DisarmingVoice());  
 }  
}

Ralts.java

import ru.ifmo.se.pokemon.\*;  
  
public class Ralts extends Pokemon {  
 public Ralts(String name, int level) {  
 super(name, level);  
 setStats(28,25,25,45,35,40);  
 setType(Type.*PSYCHIC*, Type.*FAIRY*);  
 setMove(new ShadowBall(), new ThunderWave());  
 }  
}

PokeAttacks.java

//go go brbr  
import ru.ifmo.se.pokemon.\*;

class Facade extends PhysicalMove{  
 protected Facade(){  
 super(Type.*NORMAL*, 70, 100);  
 }  
 @Override  
 protected void applyOppDamage(Pokemon def, double damage){  
 Status PokCon = def.getCondition();  
 if (PokCon.equals(Status.*BURN*) || PokCon.equals(Status.*POISON*) || PokCon.equals(Status.*PARALYZE*)) {  
 def.setMod(Stat.*HP*, (int) Math.*round*(damage) \* 2);  
 }  
 }  
 @Override  
 protected String describe(){  
 return "сила удваивается, если использующий обожжён, парализован или отравлен";  
 }  
}  
  
class ScaryFace extends PhysicalMove {  
 protected ScaryFace() {  
 super(Type.*NORMAL*, 0, 100);  
 }  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 p.setMod(Stat.*SPEED*, -2);  
 }  
 @Override  
 protected String describe(){  
 return "уменьшить скорость цели на 2 ступени";  
 }  
}  
  
class Amnesia extends PhysicalMove {  
 protected Amnesia() {  
 super(Type.*PSYCHIC*,0,0);  
 }  
 @Override  
 protected void applySelfEffects(Pokemon p) {  
 p.setMod(Stat.*SPECIAL\_DEFENSE*,2);  
 }  
 @Override  
 protected String describe(){  
 return "увеличивать особую защиту пользователя на 2 ступени";  
 }  
}  
  
class SwordsDance extends PhysicalMove {  
 protected SwordsDance() {  
 super(Type.*NORMAL*,0,0);  
 }  
 @Override  
 protected void applySelfEffects(Pokemon p) {  
 p.setMod(Stat.*ATTACK*,2);  
 }  
 @Override  
 protected String describe() {  
 return "увеличивать атаку пользователя на 2 ступени";  
 }  
}  
  
class DoubleTeam extends PhysicalMove {  
 protected DoubleTeam() {  
 super(Type.*NORMAL*,0,0);  
 }  
 @Override  
 protected void applySelfEffects(Pokemon p) {  
 p.setMod(Stat.*EVASION*,1);  
 }  
 @Override  
 protected String describe() {  
 return "увеличивать уклонение пользователя на 1 ступени";  
 }  
}  
  
class RockPolish extends PhysicalMove {  
 protected RockPolish() {  
 super(Type.*ROCK*,0,0);  
 }  
 @Override  
 protected void applySelfEffects(Pokemon p) {  
 p.setMod(Stat.*SPEED*,2);  
 }  
 @Override  
 protected String describe() {  
 return "увеличивать скорость пользователя на 2 ступени";  
 }  
}  
  
class Bulldoze extends PhysicalMove{  
 protected Bulldoze(){  
 super(Type.*GROUND*,60,100);  
 }  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 p.setMod(Stat.*SPEED*, -1);  
 }  
 @Override  
 protected String describe(){  
 return "уменьшить скорость цели на 1 ступени";  
 }  
}  
  
class Psychic extends PhysicalMove {  
 protected Psychic(){  
 super(Type.*PSYCHIC*,90,100);  
 }  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 if (Math.*random*() <= 0.1) p.setMod(Stat.*SPECIAL\_DEFENSE*,-1);  
 }  
 @Override  
 protected String describe(){  
 return "имеет 10% шанс уменьшить особую защиту цели на 1 ступени";  
 }  
}  
  
class ShadowBall extends PhysicalMove {  
 protected ShadowBall(){  
 super(Type.*GHOST*,80,100);  
 }  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 if (Math.*random*() <= 0.2) p.setMod(Stat.*SPECIAL\_DEFENSE*,-1);  
 }  
 @Override  
 protected String describe(){  
 return "имеет 20% шанс уменьшить особую защиту цели на 1 ступени";  
 }  
}  
  
class ThunderWave extends PhysicalMove{  
 protected ThunderWave() {  
 super(Type.*ELECTRIC*,0,90);  
 }  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 Effect.*paralyze*(p);  
 }  
 @Override  
 protected String describe(){  
 return "парализовать цель";  
 }  
}  
  
class DisarmingVoice extends SpecialMove{  
 protected DisarmingVoice(){  
 super(Type.*FAIRY*,40,999999999);  
 }  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 p.setMod(Stat.*ACCURACY*, 0);  
 p.setMod(Stat.*EVASION*, 0);  
 }  
 @Override  
 protected String describe(){  
 return "наносить урон напрямую";  
 }  
}  
  
class FocusBlast extends PhysicalMove {  
 protected FocusBlast(){  
 super(Type.*FIGHTING*,120,70);  
 }  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 if (Math.*random*() <= 0.1) p.setMod(Stat.*SPECIAL\_DEFENSE*,-1);  
 }  
 @Override  
 protected String describe(){  
 return "Имеет 10% шанс уменьшить особую защиту цели на 1 ступени";  
 }  
}

Результат работы

Bouffalant Travis из команды черных вступает в бой!

Ralts Kendrick из команды полосатых вступает в бой!

Ralts Kendrick парализовать цель.

Критический удар!

Bouffalant Travis теряет 6 здоровья.

Bouffalant Travis парализован

Bouffalant Travis уменьшить скорость цели на 2 ступени.

Критический удар!

Ralts Kendrick теряет 7 здоровья.

Ralts Kendrick уменьшает скорость.

Ralts Kendrick парализовать цель.

Критический удар!

Bouffalant Travis теряет 4 здоровья.

Bouffalant Travis сила удваивается, если использующий обожжён, парализован или отравлен.

Критический удар!

Ralts Kendrick имеет 20% шанс уменьшить особую защиту цели на 1 ступени.

Критический удар!

Bouffalant Travis теряет 1 здоровья.

Bouffalant Travis не замечает воздействие типа GHOST

Bouffalant Travis промахивается

Bouffalant Travis промахивается

Bouffalant Travis промахивается

Ralts Kendrick имеет 20% шанс уменьшить особую защиту цели на 1 ступени.

Критический удар!

Bouffalant Travis теряет 1 здоровья.

Bouffalant Travis не замечает воздействие типа GHOST

Ralts Kendrick парализовать цель.

Критический удар!

Ralts Kendrick теряет 8 здоровья.

Ralts Kendrick парализован

Оба покемона теряют сознание.

Kirlia Tyler из команды полосатых вступает в бой!

Bouffalant Travis уменьшить скорость цели на 2 ступени.

Критический удар!

Kirlia Tyler теряет 10 здоровья.

Kirlia Tyler уменьшает скорость.

Kirlia Tyler парализовать цель.

Критический удар!

Bouffalant Travis теряет 8 здоровья.

Bouffalant Travis теряет сознание.

Amaura Kanye из команды черных вступает в бой!

Kirlia Tyler парализовать цель.

Критический удар!

Amaura Kanye теряет 5 здоровья.

Amaura Kanye парализован

Amaura Kanye промахивается

Amaura Kanye промахивается

Amaura Kanye промахивается

Kirlia Tyler парализовать цель.

Критический удар!

Amaura Kanye теряет 5 здоровья.

Amaura Kanye уменьшить скорость цели на 1 ступени.

Критический удар!

Kirlia Tyler теряет 9 здоровья.

Kirlia Tyler уменьшает скорость.

Kirlia Tyler теряет сознание.

Gallade Abel из команды полосатых вступает в бой!

Amaura Kanye промахивается

Gallade Abel парализовать цель.

Amaura Kanye теряет 2 здоровья.

Amaura Kanye уменьшить скорость цели на 1 ступени.

Критический удар!

Gallade Abel теряет 11 здоровья.

Gallade Abel уменьшает скорость.

Gallade Abel имеет 20% шанс уменьшить особую защиту цели на 1 ступени.

Amaura Kanye теряет 5 здоровья.

Amaura Kanye промахивается

Gallade Abel имеет 20% шанс уменьшить особую защиту цели на 1 ступени.

Amaura Kanye теряет 7 здоровья.

Amaura Kanye теряет сознание.

Aurorus Frank из команды черных вступает в бой!

Gallade Abel парализовать цель.

Aurorus Frank теряет 3 здоровья.

Aurorus Frank парализован

Aurorus Frank промахивается

Aurorus Frank промахивается

Gallade Abel Имеет 10% шанс уменьшить особую защиту цели на 1 ступени.

Aurorus Frank теряет 47 здоровья.

Aurorus Frank теряет сознание.

В команде черных не осталось покемонов.

Команда полосатых побеждает в этом бою!

Вывод:

В процессе выполнения лабораторной работы были получены навыки использования бъектно-ориентированного подхода программирования при использовании языка Java.